UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/998,755	12/03/2001	Jeff L. Hunter	TI-31599	6452

23494	7590	09/01/2004	

TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

EXAMINER
YIGDALL, MICHAEL J

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 09/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	09/998,755	HUNTER ET AL.
	Examiner	Art Unit	
	Michael J. Yigdall	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _03 December 2001_.
2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-13_ is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) _1-13_ is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☒ The drawing(s) filed on _03 December 2001_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a)☐ All   b)☐ Some * c)☐ None of:
      1.☐ Certified copies of the priority documents have been received.
      2.☐ Certified copies of the priority documents have been received in Application No. _____.
      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
        application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)
2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.
4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .
5)☐ Notice of Informal Patent Application (PTO-152)
6)☐ Other: _____.

DETAILED ACTION

1.   Claims 1-13 are pending and have been examined. The priority date considered for the application is January 24, 2001.

Information Disclosure Statement

2.   The listing of references in the specification is not a proper information disclosure statement (see for example the patents referenced on pages 7 and 11). 37 CFR 1.98(b) requires a list of all patents, publications, or other information submitted for consideration by the Office, and MPEP § 609 A(1) states, "the list may not be incorporated into the specification but must be submitted in a separate paper." Therefore, unless the references have been cited by the examiner on form PTO-892, they have not been considered.

Double Patenting

3.   The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground

provided the conflicting application or patent is shown to be commonly owned with this

application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal

disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37

CFR 3.73(b).

4.   Claims 1-13 are provisionally rejected under the judicially created doctrine of

obviousness-type double patenting as being unpatentable over claims 1-10 of copending

Application No. 09/998,330. Although the conflicting claims are not identical, they are not

patentably distinct from each other because both recite analogous methods and systems for

transparently writing to shared memory when debugging a multiple processor system.

This is a provisional obviousness-type double patenting rejection because the conflicting

claims have not in fact been patented.

5.   Claims 1-13 are provisionally rejected under the judicially created doctrine of

obviousness-type double patenting as being unpatentable over claims 1-19 of copending

Application No. 09/998,329. Although the conflicting claims are not identical, they are not

patentably distinct from each other because both recite analogous methods and software

development systems for transparently writing to shared memory when debugging a multiple

processor system.

This is a provisional obviousness-type double patenting rejection because the conflicting

claims have not in fact been patented.

6.   Claims 1-13 are provisionally rejected under the judicially created doctrine of
obviousness-type double patenting as being unpatentable over claims 1-24 of copending
Application No. 09/998,756. Although the conflicting claims are not identical, they are not
patentably distinct from each other because both recite analogous methods and systems for
maintaining coherency of software breakpoints in shared memory when debugging a multiple
processor system.

        This is a provisional obviousness-type double patenting rejection because the conflicting
claims have not in fact been patented.

Claim Rejections - 35 USC § 112

7.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the
> subject matter which the applicant regards as his invention.

8.   Claims 6 and 7 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite
for failing to particularly point out and distinctly claim the subject matter which applicant
regards as the invention.

        Claim 6 recites "a third plurality of processors." There is insufficient antecedent basis for
this limitation in the claim. Claim 1, upon which claim 6 depends, recites only one plurality of
processors. There is no second plurality of processors recited in the claim to provide a basis for
a third plurality of processors.

        Moreover, claim 6 recites "the new instruction" and "the old instruction." There is
insufficient antecedent basis for this limitation in the claim. Claim 1, upon which claim 6
depends, does not recite such instructions.

Claim 7 is indefinite because it depends upon claim 6.

Claim Rejections - 35 USC § 103

9.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

10.   Claims 1-4 and 6-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S.

Pat. No. 6,088,770 to Tarui et al. (hereinafter "Tarui") in view of U.S. Pat. No. 6,065,078 to

Falik et al. (hereinafter "Falik").

With respect to claim 1, Tarui discloses a method for transparently writing to shared

memory in a multiple processor system (see the abstract), but does not expressly disclose

debugging the multiple processor system.

However, Falik discloses debugging a multiple processor system (see the abstract). It

would have been obvious to one of ordinary skill in the art at the time the invention was made to

employ the method of Tarui when debugging the multiple processor system, as taught by Falik.

Such a combination would serve to reduce congestion and access latency (see Tarui, column 19,

lines 12-14) when debugging software executing on the plurality of processors (see Falik,

column 1, lines 46-49).

Tarui in view of Falik further discloses the steps of:

(a) creating a software memory map of the memory usage of a plurality of processors in the system to be debugged (see Tarui, column 9, lines 12-31, which shows memory configuration information, i.e. a memory map, for a plurality of nodes or processors);

(b) activating a first debug session associated with a first processor of the plurality of processors and at least a second debug session associated with a second processor of the plurality of processors (see Falik, column 2, lines 37-43, which shows activating a debug session for each processor of the plurality of processors);

(c) detecting a write request to a shared memory location by the first debug session (see Tarui, column 9, lines 1-11, which shows detecting an access request, e.g. a write request, to a shared memory location);

(d) if the first processor associated with the first debug session has write access to the shared memory location then selecting the first processor to perform the write request (see Tarui, column 9, lines 42-47, which shows determining the processor having access to the shared memory location, and column 14, lines 22-26, which shows selecting the internal processor, i.e. the first processor, to perform the write request), else performing the following steps:

(i) searching the software memory map to determine if the second processor has write access to the shared memory location (see Tarui, column 9, lines 32-41, which shows determining if a remote processor, i.e. a second processor, has access to the shared memory location, based on searching the memory map);

(ii) selecting the second processor to perform the write request (see Tarui, column 15, lines 46-52, which shows selecting the second processor to perform the write request); and

(e) passing the write request initiated by the first debug session to the selected processor

for execution (see Tarui, column 14, lines 22-26 and column 15, lines 46-52, which shows

passing the write request to the selected processor).

With respect to claim 2, Tarui in view of Falik further discloses the limitation wherein the

step of passing the write request comprises the steps of:

(a) searching the software memory map for a second plurality of processors that have

read access to the shared memory location (see Tarui, column 6, line 66 to column 7, line 9,

which shows searching a memory table or map for a plurality of processors that have access to

the shared memory location);

(b) broadcasting the write request to the second plurality of processors (see Tarui, column

10, line 66 to column 11, line 5, which shows broadcasting the request to the plurality of

processors); and

(c) performing cache coherency updates in response to the write request in each of the

second plurality of processors (see Tarui, column 6, lines 46-53, which shows issuing and

executing CCC commands, and column 3, lines 5-11, which shows that the CCC commands are

for performing cache coherency updates).

With respect to claim 3, Tarui in view of Falik further discloses the limitation wherein the

step of broadcasting the write request comprises indicating that the write request is intended for

maintaining cache coherency as opposed to a normal write request (see Tarui, column 14, lines

5-14, which shows broadcasting an I command, and column 8, lines 7-11, which shows that the I

command is for maintaining cache coherency as opposed to a normal write request).

With respect to claim 4, Tarui in view of Falik further discloses the limitation wherein the step of performing comprises using cache coherency capabilities, if any, of a processor in response to the write request intended for maintaining cache coherency (see Tarui, column 8, lines 7-11, which shows using a cache coherency capability in response to a write request).

With respect to claim 6, Tarui in view of Falik further discloses the limitation wherein the step of passing the write request comprises:

(a) determining that the write request is to a shared memory location at which a software breakpoint has been set (see Falik, column 16, lines 36-45, which shows determining a location at which a breakpoint has been set);

(b) searching the software memory map to find a third plurality of processors with read access to the shared memory location (see Tarui, column 6, line 66 to column 7, line 9, which shows searching a memory table or map for a plurality of processors that have access to the shared memory location);

(c) sending the third plurality of processors the new instruction for the shared memory location (see Tarui, column 10, line 66 to column 11, line 5, which shows broadcasting the command or instruction to the plurality of processors);

(d) updating a software representation maintained for software breakpoints for each of the third plurality of processors to replace the old instruction with the new instruction (see Falik, column 16, lines 36-45, which shows updating the breakpoint representation for each of the plurality of processors); and

(e) resetting the software breakpoint with the new instruction in the write request (see

Tarui, column 6, lines 46-53, which shows issuing and executing CCC commands, and column

3, lines 5-11, which shows that the CCC commands are for performing cache coherency updates;

note that when applied to a memory location at which a breakpoint has been set, a cache

coherency update would comprise resetting the breakpoint).

With respect to claim 7, Tarui in view of Falik further discloses the limitation wherein the

step of resetting comprises:

(a) clearing the software breakpoint at the shared memory location (see Falik, column 17,

lines 27-44, which shows clearing an abort bit associated with a breakpoint);

(b) performing the write request (see Tarui, column 14, lines 22-26 and column 15, lines

46-52, which shows passing the write request to the selected processor); and

(c) setting a software breakpoint at the shared memory location (see Falik, column 16,

lines 45-50, which shows setting an abort bit associated with a breakpoint).

With respect to claim 8, Tarui in view of Falik further discloses the limitation wherein the

steps of the method are performed iteratively to write to successive shared memory locations

(note that the steps of the method are inherently performed iteratively when writing to successive

shared memory locations).

With respect to claim 9, Tarui in view of Falik further discloses the step of reading the

shared memory location by the first processor after the contents of the shared memory location

have been changed by the write request (see column 8, lines 1-6 and 37-42, which shows reading

the shared memory location when the contents have been changed and invalidated by another

processor, e.g. because of the write request).

With respect to claim 10, the limitations recited in the claim are analogous to the

limitations of claim 1 (see the rationale applied to claim 1 above).

Tarui in view of Falik further discloses a software development system, comprising:

(a) a memory storage system holding a software development tool program (see Falik,

column 2, lines 37-43, which shows a software development tool program);

(b) a host computer connected to the memory storage system, the host computer operable

to execute the software development tool program (see Falik, column 2, lines 47-51, which

shows a host computer for executing the software development tool program);

(c) a test port for connecting to a hardware system (see Falik, column 3, lines 17-23,

which shows a test access port for connecting to a hardware system), the hardware system being

comprised of multiple processors with shared memory and operable to execute an application

program (see Falik, column 2, lines 51-55, which shows multiple processors and memory, and

column 2, line 66 to column 3, line 3, which shows that the memory may be shared memory).

With respect to claim 11, the limitations recited in the claim are analogous to the

limitations of claim 2 (see the rationale applied to claim 2 above).

With respect to claim 12, the limitations recited in the claim are analogous to the

limitations of claim 10 (see the rationale applied to claim 10 above).

With respect to claim 13, the limitations recited in the claim are analogous to the limitations of claim 11 (see the rationale applied to claim 11 above).

11.   Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Tarui in view of Falik, as applied to claim 2 above, and further in view of U.S. Pat. No. 6,539,500 to Kahle et al. (hereinafter "Kahle").

With respect to claim 5, although Tarui in view of Falik discloses a method for transparently writing to shared memory when debugging a multiple processor system (see above), Tarui in view of Falik does not expressly disclose the limitations wherein:

(a) the step of creating comprises denoting in the software memory map the shared memory locations that contain program instructions;

(b) the step of passing the write request additionally comprises the step of determining that the shared memory location contains a program instruction; and

(c) the cache is an instruction cache.

However, Kahle discloses instruction tracing in shared memory multiprocessor system (see the abstract). Instructions may be traced and analyzed at native speeds, simultaneously with instruction execution (see column 2, lines 5-20).

Kahle further discloses, as in part (a) above, denoting in a memory map the address space or locations of program instructions in shared memory (see column 3, lines 19-27).

Kahle further discloses, as in part (b) above, determining that the shared memory location contains a program instruction (see column 3, lines 48-50).

Kahle further discloses, as in part (c) above, an instruction cache (see step 400 in FIG. 4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to extend the debugging of Tarui and Falik with the shared memory instruction tracing features taught by Kahle, as presented above, for the purpose of reducing the time necessary to debug the multiple processor system (see Kahle, column 1, lines 45-55).

Conclusion

12.   The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure. U.S. Pat. No. 6,295,584 to DeSota et al. discloses a method for transparently accessing shared memory in a multiprocessor system comprising a memory map.

13.   Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (703) 305-0352. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).
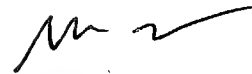
MY

Michael J. Yigdall
Examiner
Art Unit 2122

mjy

WEI Y. ZHEN
PRIMARY EXAMINER